Clouds Course

Project Report

Student Name and email:

Adib Rachid - rachid@eurecom.fr

Your project GitHub link (containing source code):

https://github.com/adibrachid1/Covid19Data

Your hosted website link on Google cloud:

https://covid19-c4432.web.app/

1) **Beside Google Cloud Platform, what are the frameworks/libraries you used to develop your application? (i.e. Angular, Bootstrap, FontAwesome, React, ng2-charts…)**

Main ones:

Angular: web development

Bootstrap: Enhancing style

Ng2-charts: Charts

Material-select: drop down country list

2) What are the API calls that you used to get the data from the COVID19 API? (i.e. https://api.covid19api.com/summary)

https://api.covid19api.com/**summary**

=> Get all information for all countries for the table.
=> Get the summary for the world section page (if not up to date in DB).
=> Get the summary for the country section page (if not up to date in DB).

https://api.covid19api.com/**world**?from='*from*'&to=*to*'

=> Get data to produce the 7 days chart and the chart for all days from April 2020. (direct find in the response)

https://api.covid19api.com/**total/country**/'*countrySlug*'?from='*from*'&to='*to*

=> Get data to produce the 7 days chart. (not direct in the response), by calculating it with the total number of day x minus the total number of day x-1.

https://api.covid19api.com/**countries**

=> Get data to find specific country if exists while navigating directly to a country.

3) Briefly explain how you implemented the authentication functionality to allow a user to sign in

Similarly to the Expenses application, after importing firebase and add necessary variables, 'sign In with pop up' method of the angular fire auth is called and the google auth provider is passed to it. We store the credentials while setting the function to asynchronous. The credentials are also stored in a User object containing the 3 main information required which are his username to be shown when signed in, the UID to keep it and the email and the credentials will also be saved in the fire store with the set method to save data there. After sign in the user will be navigated to the home page. The database will only be updated at first sign in to store data. We could also add the date as last log in to get more useful data. I also stored the user in the local storage to keep track of the logged in user while navigating and refreshing pages with the help of the JSON stringify to store it in the local storage of the browser. I also adjusted the get user information method to maintain the user data across many navigations. I also added guards so that no one can navigate internally if not logged in.

4) Briefly explain how you implemented the news feature to allow only eligible users to add news

Along the other collections, I created a super users collection where I add inside of it the Gmail emails of the users which are eligible to post news. Then in the home and country pages, I have 2 main divisions for the news. One to add and one containing the news data.
The 'Add news' division have a condition with the '*ngIf' with a Boolean condition to show the division which is if the user is a super user which will be checked at first on page load. If the user is a super user, the condition will be false which will not show the division to add a news. To enhance the news features, we could add a delete button in front of all news set by the user itself. I have worked on Gmail email account because it wouldn't make sense to work on the UID of the Gmail since when we want to add or remove someone, we cannot easily identify him with the UID and also this won't create any security issues.

5) Which part of this project you consider it more difficult than other parts?

Parts that were more difficult than others:

- API source is very bad: some data are inconsistent, not all answers can be directly found in APIS. Should do some computations to find them like the 7 days chart for each country where we must get 8 days and subtract each 2 days to find the daily numbers from total.

- Find correct APIs to use for each task

- Handle data to only update when data is not updated the same day

- Have the same exact design as the instructions

- All other parts were simple

- Some problems with managing which update for which country because things were messing up between each other but this was solved by sending extra information between functions

6) What did you find interesting in Google Cloud Platform?

The most interesting part in GCP is that it enables a developer to develop whatever he wants while offering him for free all needed tools: storage, real time database, database, hosting, ML tools. And of course, very easy ways to access and use these features like accessing and modifying the Cloud firestore and the authentication feature which is ready to be used in a very easy way to be integrated in our application and all the management is done by the GCP.